

고급C프로그래밍 preview

지능기전공학부 스마트기기공학전공 3학년 정건희
2023-1

1-1. 고급C프로그래밍에서 배울 것들

문자열

구조체

동적 메모리 할당

파일 입출력

2-1. Recall) stdio.h

```
#include<stdio.h>
```

printf();

표준출력(stdout) 함수

scanf();

표준입력(stdin) 함수

fprintf();

파일에 출력하는 함수(고급c-파일 입출력)

fscanf();

파일에서 입력받는 함수(고급c-파일 입출력)

getchar()

문자1개 표준입력으로 받는 함수

putchar()

문자1개를 표준출력하는 함수

gets()

문자열을 표준입력으로 받는 함수(고급c-문자열)

puts()

문자열을 표준출력하는 함수(고급c-문자열)

fopen();

파일을 여는 함수(고급c-파일 입출력)

fclose();

파일을 닫는 함수(고급c-파일 입출력)

2-2. 문자열 헤더

```
#include<string.h>
```

strcpy()

문자열을 복사하는 함수

strcat()

문자열을 이어붙이는 함수

strcmp()

문자열을 비교하는 함수

strlen()

문자열의 길이를 반환하는 함수 (int형으로 반환)

2-3. 문자열이란?

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<string.h>

int main() {

    char sejong[100];
    scanf("%s", &sejong);
    return 0;
}
```



2-3. 문자열이란?

조사식 1

검색(Ctrl+E) 🔍 < > 검색 심도: 3

이름	값	형식
sejong	0x00000017d92ff830 "sejong"	char[100]
[0]	115 's'	char
[1]	101 'e'	char
[2]	106 'j'	char
[3]	111 'o'	char
[4]	110 'n'	char
[5]	103 'g'	char
[6]	0 '\0'	char
[7]	-52 '?'	char
[8]	-52 '?'	char

sejong[0] sejong[1] ...

s	e	j	o	n	g	NULL			
---	---	---	---	---	---	------	--	--	--

NULL 문자는 문자열의 끝을 나타내는 특수한 문자 => 문자열에는 없어서 안될 존재

2-3. 문자열이란?

조사식 1

검색(Ctrl+E) 🔍 < > 검색 심도: 3

이름	값	형식
sejong	0x00000017d92ff830 "sejong"	char[100]
[0]	115 's'	char
[1]	101 'e'	char
[2]	106 'j'	char
[3]	111 'o'	char
[4]	110 'n'	char
[5]	103 'g'	char
[6]	0 '\0'	char
[7]	-52 '?'	char
[8]	-52 '?'	char

Microsoft Visual Studio 디버그 콘솔

```
sejong
sejong
C:\Users\MisterJerry\source\repos\coding_mentor\coding_mentor\Debug\coding_mentor.exe(프로세스 13000개)이(가) 종료되었습니다
(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

`printf("%s", sejong);` → NULL문자가 나오기 전까지 출력

2-4. 문자열 함수들

```
#include<string.h>
```

strcpy()

문자열을 복사하는 함수

strcat()

문자열을 이어붙이는 함수

strcmp()

문자열을 비교하는 함수

strlen()

문자열의 길이를 반환하는 함수 (int형으로 반환)

위의 함수를 사용하기 위해서는 NULL문자의 개념을 알아야 한다.

3-1. 구조체란?

int id[10]; //학생들의 학번을 저장하는 배열

101	102	103	104	105	106	107	108	109	110
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

int math[10]; //학생들의 수학점수를 저장하는 배열

95	97	48	68	25	34	75	23	75	56
----	----	----	----	----	----	----	----	----	----

int kor[10]; //학생들의 국어점수를 저장하는 배열

65	34	54	74	86	34	67	86	76	45
----	----	----	----	----	----	----	----	----	----

뭉어서 하나로 관리하면 편할 것 같은데...?

3-1. 구조체란?

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<string.h>
```

```
struct student {
    int id;
    int math;
    int kor;
};
```

멤버변수

```
int main() {
    struct student st[10];
    return 0;
}
```

구조체

id[0] id[1] id[2]

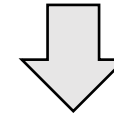
101	102	103	104	105	106	107	108	109	110
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

math[0] math[1] math[2]

95	97	48	68	25	34	75	23	75	56
----	----	----	----	----	----	----	----	----	----

kor[0] kor[1] kor[2]

65	34	54	74	86	34	67	86	76	45
----	----	----	----	----	----	----	----	----	----



st[0]

st[1]

st[2]

101	95	65	102	97	34	103	48	54	...
-----	----	----	-----	----	----	-----	----	----	-----

st[0].id st[0].math st[0].kor st[1].id st[1].math st[1].kor st[2].id st[2].math st[2].kor

구조체의 멤버변수는 구조체 멤버 연산자(.)를 통해서 접근

4-1. 동적할당 관련 함수들

```
#include<stdlib.h>
```

malloc()

동적으로 메모리 할당 함수

calloc()

동적으로 메모리 할당 후 초기화 시켜주는 함수

realloc()

할당된 메모리의 크기를 변화시켜주는 함수

4-2. 정적할당과 동적할당?

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>

int main() {
    int arr[10];
    return 0;
}
```

메모리에 10칸짜리 배열을 정적으로 할당

- 선언만 하면 사용가능(So simple)
- 프로그램이 종료될 때 까지 arr[10]이 메모리를 차지함
- 배열의 크기를 바꿀 수 없음

4-2. 정적할당과 동적할당?

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<stdlib.h>

int main() {
    int* arr;
    arr = (int*)malloc(10* sizeof(int));

    return 0;
}
```

메모리에 10칸짜리 배열을 동적으로 할당 (직접 크기 지정)

- 원하는 만큼만 할당할 수 있어서 메모리를 효율적으로 관리 가능
- 원할 때 free()를 통해서 메모리에서 해제할 수 있음
- 할당 된 크기를 바꿀 수 있음(realloc)
- 메모리 누수가 생길 수 있음

5-1. 파일 입출력

말 그대로 직접 파일을 읽어와서 입력을 받을 수 있고(read)
파일에 직접 출력(write)할 수 있음

5-2. 파일 입출력-fopen()

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
```

```
int main() {
```

```
FILE* fp;
```

파일 포인터 : 파일의 위치(주소)를 가리키는 함수

```
fp = fopen("mentoring.txt", "r");
```

fopen() : 파일을 여는 함수로 mentoring.txt파일을 r모드(read)로 파일을 열겠다는 뜻

```
return 0;
```

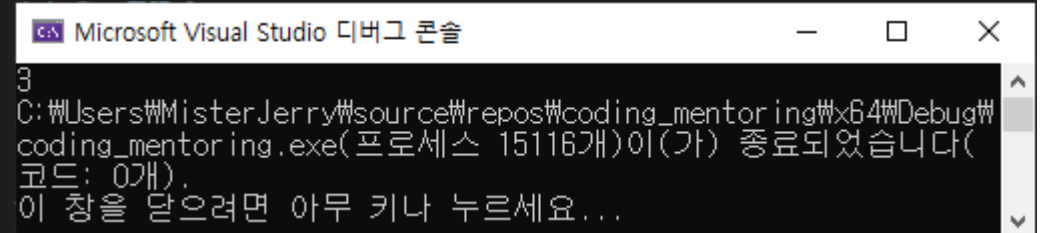
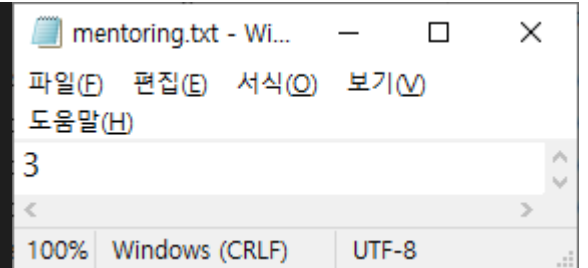
```
}
```

5-3. 파일 입출력-fscanf()

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
int main() {
    FILE* fp;
    int a;
    fp = fopen("mentoring.txt", "r");
    fscanf(fp, "%d", &a);
    printf("%d", a);

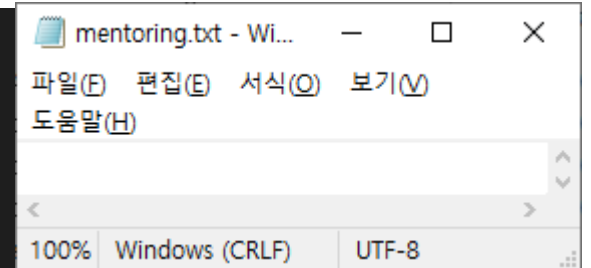
    return 0;
}
```

fp(mentoring.txt)에서 숫자1개를 읽어서 a에 저장 (scanf와 유사)

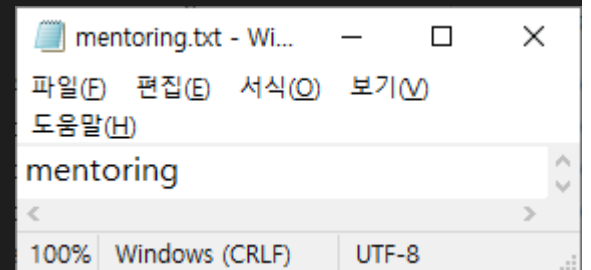


5-4. 파일 입출력-fprintf()

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
int main() {
    FILE* fp;
    int a;    write하기 위해서는 파일을 w모드(write)모드로 열기
    fp = fopen("mentoring.txt", "w");
    fprintf(fp, "%s", "mentoring");
    fp(mentoring.txt)에 문자열 출력 (printf와 유사)
    return 0;
}
```



<before>



<after>

6. QnA

A large, bold, black text "Q&A" centered within a light gray rounded rectangular box.

Q&A